数理解析研究所講究録
646巻 1988年 136-146

一般逆行列の数式処理的解法について

愛媛大学・工　野田松太郎（Matu-Tarow NODA）
愛媛大学・工　越智　正明（Masaaki OCHI）

<u>1．はじめに</u>　一般逆行列がMooreにより提案されて半世紀近くになる。その利用は次のような問題に重要である。m個のデータ点をn個の基底関数の一次結合（多項式基底等）で表す問題。$m=n$なら、連立一次方程式を解く単純な問題に帰結されるが、$m>n$の場合にはデータ過多で、データ点で一次従属になるような基底が含まれる。また、連立一次方程式の係数行列は悪条件となり、LU分解法等の通常の数値計算法で解を得るのは困難な場合等である。もちろん、単に連立方程式を解くことだけを考えるならあえて一般逆行列を求める必要はないが、逆行列そのものの挙動などを研究する場合には正確な逆行列の計算は必須である。現在の一般逆行列の計算は特異値分解法による数値計算によって行われるのが大半である。この方法では極端に小さな特異値による演算を避けるためしきい値を設ける。小さな特異値が浮動小数点演算の誤差の影響による場合は、しきい値の設定は良い結果を与える。一方、小さな特異値が意味ある場合、しきい値操作は本来独立であるべきベクトルを従属関係にあるとみなし値

域を縮小することに対応し、結果は信用できなくなる。結果は常に誤差の不安に脅かされ続ける。そこで、誤差なしに一般逆行列を求める手法の確立が要求される。Gregory[1]等は数値計算とは異なった立場から、モジュラ演算を用いて整数体上のみで誤差のない一般逆行列の計算を行おうとしている。一方、誤差のない計算を実現できるシステムとして数式処理システムを考えることは必然と思われるが、その試みはほとんどなく、わずかに数式処理システムMACSYMA上に実現されたもの[2]が発表されているのみと思われる。本論では数式処理システム上へ数式処理に適すると思われる各種のアルゴリズムのインプリメントを2で行い、3で計算量、メモリー効率の視点からそれらの評価、検討を行う。具体例はMACSYMA、REDUCE上で行なった。本論のまとめは4にある。

2．一般逆行列を求めるアルゴリズム　数式処理に適したアルゴリズムとして次の4種の直接法のアルゴリズムを考える。これらの詳細は文献1にある。

① Hermiteのアルゴリズム

任意のm×n行列Aに対し、一般逆行列に関する等式

$$A^{+}=A^{T}M_{R}^{-}AA^{T} \qquad (3.1)$$

を基に一般逆行列を計算する。ここでM_{R}^{-}は行列Mの反射型一般逆行列を表わし、$M=(AA^{T})^{2}$である。アルゴリズム

の主要部はM_R^-の計算にある。行列Aの rank をrとすると、

$$FM^TE^T=R$$
$$=\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \qquad (3.2)$$

を満たす正則行列E，Fが存在し、一般逆行列の性質より

$$M_R^-=F^TRE \qquad (3.3)$$

となる。m×2m行列[M：I]への行基本操作E_iにより、

$$E_s\cdots E_2E_1[M:I]=[M_1:E] \qquad (3.4)$$

とEを求める。同様に行列[M_1^T：I]への行基本操作より、F，Rを求める。すなわち、

$$F_s\cdots F_2F_1[M_1^T:I]=[R:F]. \qquad (3.5)$$

A^+は（3.3)、（3.1）式より得られる。

Hermite のアルゴリズムはインプリメントが容易で rank の決定もできるが、複雑なピボット操作の要がある。

② <u>Albert の極限形式</u>

aをスカラー量とし、

$$a^+=\begin{cases} a^{-1} & (a\neq 0\text{のとき}) \\ 0 & (a=0\text{のとき}) \end{cases} \qquad (3.6)$$

とすると、a^+はMoore-Penrose 型一般逆行列の定義を満たす。a→0のときを見れば、一般逆行列は連続関数ではないことは自明である。aを列ベクトルとしたときの一般化は

$$a^+ = \begin{cases} a^T / a^T a & (a \neq 0 \text{のとき}) \\ 0^T & (a = 0 \text{のとき}) \end{cases} \qquad (3.7)$$

となる。Aを任意の行列とすると一般逆行列A^+は次のいずれかになる。これをAlbertの極限形式という。

$$A^+ = \lim_{x \to 0} A^T (AA^T + x^2 I)^{-1} \qquad (3.8)$$

$$A^+ = \lim_{x \to 0} (A^T A + x^2 I)^{-1} A^T \qquad (3.8')$$

文献2でインプリメントされたものはAlbertの極限形式であるが、変数xを含む行列計算による必要メモリ量の増大と極限計算の煩雑さを要する。

③ Grevilleのアルゴリズム

入力行列を列分割し、一列ずつ取り込み、既に計算した下位行列の一般逆行列と新しく取り込んだ列とから新しい行列の一般逆行列を計算する手続きを再帰的に繰り返す数値算用に開発された再帰的アルゴリズムである。m×n行列Aを

$$A = [a_1^T, a_2^T, \ldots, a_n^T]^T$$

と行ベクトルa_iに分割し、i×n行列A_iを

$$A_1 = a_1$$

$$A_i = \left[\begin{array}{c} A_{i-1} \\ \hline a_i^T \end{array} \right] \qquad (3.9)$$

と定める。 $n \times i$ 行列 $A_i^\dagger$ は次式を順次計算する。

$$A_i^\dagger = \left[\begin{array}{c|c} A_{i-1}^\dagger - d_i b & b_i^T \end{array} \right] \tag{3.10}$$

ここで d_i, b_i および c_i はおのおの

$$d_i = a_i A_{i-1}^\dagger \tag{3.11}$$

$$b_i = \begin{cases} \dfrac{c_i}{c_i c_i^T} & (c_i = 0) \\ \dfrac{d_i (A_{i-1}^\dagger)^T}{1 + d_i d_i^T} & (c_i \neq 0) \end{cases} \tag{3.12}$$

$$c_i = a_i - d_i A_{i-1} \tag{3.13}$$

である。また、初項 $A_1^\dagger$ は

$$A_1^\dagger = \begin{cases} \dfrac{a_1^T}{a_1 a_1^T} & (a_1 \neq 0) \\ a_1^T & (a_1 = 0) \end{cases} \tag{3.14}$$

である。この手順を $i = n$ まで反復し、その時の $A_i^\dagger$ が A の Moore-Penrose型一般逆行列 A^+ となる。

再帰的手法の実現上の問題と浮動小数点計算での0の判定の困難さにより、数値計算の分野では広く用いられてはいない。しかし、数式処理ではある量が0であるか否かの判定は厳密に行え、再帰的手法の実現も容易である。

④ Decell-Leverrier のアルゴリズム

m×n複素行列Aの特性多項式に関するCayley-Hamiltonの定理の拡張に基ずく。A^HをAのHermite共役行列とすると、行列$B=AA^H$の持つ特性多項式は$a_0=1$として

$$B(\lambda)=(-1)^m(a_0\lambda^m+a_1\lambda^{m-1}+\cdots+a_m) \quad (3.15)$$

rを$a_r\neq 0$を満たす最大整数とすると、行列Aの一般逆行列は$\mathrm{rank}(A)=r$として、次のように求める。

$$A^+=\begin{cases} -\dfrac{1}{a_r}A^H(a_0B^{r-1}+a_1B^{r-2}+\cdots+a_{r-1}I) & (r>0) \\ 0 & (r=0) \end{cases} \quad (3.16)$$

a_r及びA^+を効率的に求めるために次の計算をする。

$$\begin{array}{lll} A_0=0 & q_0=-1 & B_0=I \\ A_1=AA^T & q_1=\mathrm{tr}A_1 & B_1=A_1-q_1I \\ A_2=A_1B_1 & q_2=\mathrm{tr}A_2/2 & B_2=A_2-q_2I \\ \cdots & & \\ A_r=A_1B_{r-1} & q_r=\mathrm{tr}A_r/r & B_r=A_r-q_rI \end{array} \quad (3.17)$$

ここで$q_i=-a_i$である。(3.16)式により、$r>0$のときA^+は次のようになる。

$$A^+=\frac{1}{q_r}A^TB_{r-1} \quad (3.18)$$

ただし、rは

$$A_1B_r=0 \quad (3.19)$$

が成立するまで上記のアルゴリズムを反復し決定する。

ここでも０の判定の必要性があり、すうしき処理では有効である可能性がある。

３．アルゴリズムの比較　アルゴリズムの比較を行うのに、数式処理計算では誤差の問題を考慮する必要が無いので、その演算回数すなわち計算量とメモリ効率を考えるだけで良い。演算回数として、数式処理計算では各種演算の間の計算時間の差はあまり無いので、次の量を尺度として用いる

（乗除算の回数＋加減算の回数）／２

アルゴリズム$\mathcal{A}$の演算量の主要項を$\phi(\mathcal{A})$とする。$\mathcal{A}$は、H（Hermite）、A（Albert の極限形式）、G（Greville）及びD（Decell-Leverrier）のいずれかである。入力行列は$m \times n$かつ$m, n \gg 1$。主要項$\phi(\mathcal{A})$は次のように求まる。ただし r は行列の rank である。

$$\phi(H) = m^3 + 3m^2n + 5rm^2 - 3r^2m + 2r^3/3 + (\text{ピボット操作})$$

$$\phi(A) = n^3 + 2mn^2 + (\text{極限操作})$$

$$\phi(G) = 2mn^2 + (\text{行列の取り込み操作})$$

$$\phi(D) = rm^3 - 2m^3 + 2m^2n + rm^2/2$$

長方行列の一般逆行列を求める場合に注意すべき点は明らかなように、$m < n$の場合、Albert、Greville の両アルゴリズ

ムは入力行列を転置してから計算した方が以後の計算量が少なくてすむが、他のアルゴリズムはその必要はない。以下では演算量の比較を $m<n$ の場合に有利な場合に統一する。$\phi(A)$、$\phi(G)$ は次のように書き直される。

$\phi(A)=m^3+2m^2n+$（極限操作）

$\phi(G)=2m^2n+$（行列の取り込み操作）

この式から分かる通り、極端な長方行列や、rank＝1の特別な場合を除いて Greville のアルゴリズムが演算量の面では優れているといえる。なお、ここで無視した各種の行列操作等はアルゴリズムをインプリメントする数式処理システムの機能などに大幅に依存する。

次に各アルゴリズムをメモリ効率の面から比較する。アルゴリズム $\mathcal{A}$ の実行に際し、必要最小メモリ量を $\rho(\mathcal{A})$ とする。$m\times n$ 入力行列に対しこれは次のようになる。

$\rho(H)=4m^2+2mn$

$\rho(A)=n^2+2mn$

$\rho(G)=n^2+3mn+4m-n$

$\rho(D)=5m^2+2mn$

このように、Albert、Greville の両アルゴリズム等が優れているが、Albertの極限形式では入力行列が常数のみの数値行列であっても記号変数 x を含んだ演算が必要となり、上の評

価以上のメモリ効率の低下を招く。従って、Grevilleのアルゴリズムがメモリ効率の面からは最適であるといえる。

以上のように、演算量とメモリ効率の両面から一般逆行列を求めるアルゴリズムとして、Grevilleのアルゴリズムが優れていると結論づけられる。

上の検討結果を具体的な問題に対する実行例からも調べる。実行は DEC-MICRO/VAX II 上で稼働する数式処理システム MACSYMA上でなされた。対象とする入力行列は次の３例。

1. full rank の n × n 行列：$a_{ij} = 2 * \min(i, j)$ を要素とする比較的悪条件の行列（Givens 行列）。
2. 各要素が Givens 行列と同じで２n × n の長方行列。
3. 各要素が $a_{ij} = i + j - 1$ の n × n 正方行列（rank 2）。

上の行列の例１の実行時間の結果が図１に示される。又、例２の結果も例１とほぼ同様である。例１では演算量の評価がそのまま反映され、Greville のアルゴリズムが実行時間最小であり、メモリ効率の面でも優れている。Albert の極限形式はメモリ効率が極端に悪い。例３の計算時間が図２にある。ここでは、他の場合と異なり必ずしも Greville の方法が最適ではなく、Hermite のものが僅かに優っている。これは行列の取り込み、ピボット操作等の数式処理システム固有の性質によると思われる。なお、メモリ容量６４０KBのパソコン

上での REDUCE によっても図1、２と同様の振舞いが見られる。Greville のアルゴリズムによるとパソコン上でも４０次の行列の一般逆行列の計算も可能である点を付記する。

4. むすび　以上、本論では数式処理的に一般逆行列を求める方法を比較、検討し、演算量とメモリ効率の両面から詳しく調べた。汎用的に色々な入力行列に対応するためには、アルゴリズムの解析といくつかの実行例とから Greville のアルゴリズムが最適であるという結論を得る。文献２で述べられた Albert の極限形式は、例え極限操作を省略してみても、演算量とメモリ効率の両面で有効なアルゴリズムとは思えない。さらに高速に一般逆行列を数式処理的に求めるためには、数式処理システムがより適切で有効な内部命令を付加することが必要である。

数式処理を有効にし要すると数値計算の分野では捨てられていたアルゴリズムを再発見することが可能である。誤差の混入しない計算の特質を用いるとより新しいアルゴリズム開発を考え得るであろう。

参考文献　1) R.T.Gregory & E.V.Krishnamurthy: "Methods and Applications of Error-Free Computation", Springer -Verlag, 1984.
2) W.J.Frawley: Computer Generation of Symbolic Generalized Inverses and Applications to Physics and Data Analysis, in "Applications of Computer Algebra" ed. by R.Pavelle, Kluwer Academic Pub., 1985, pp.415-426.

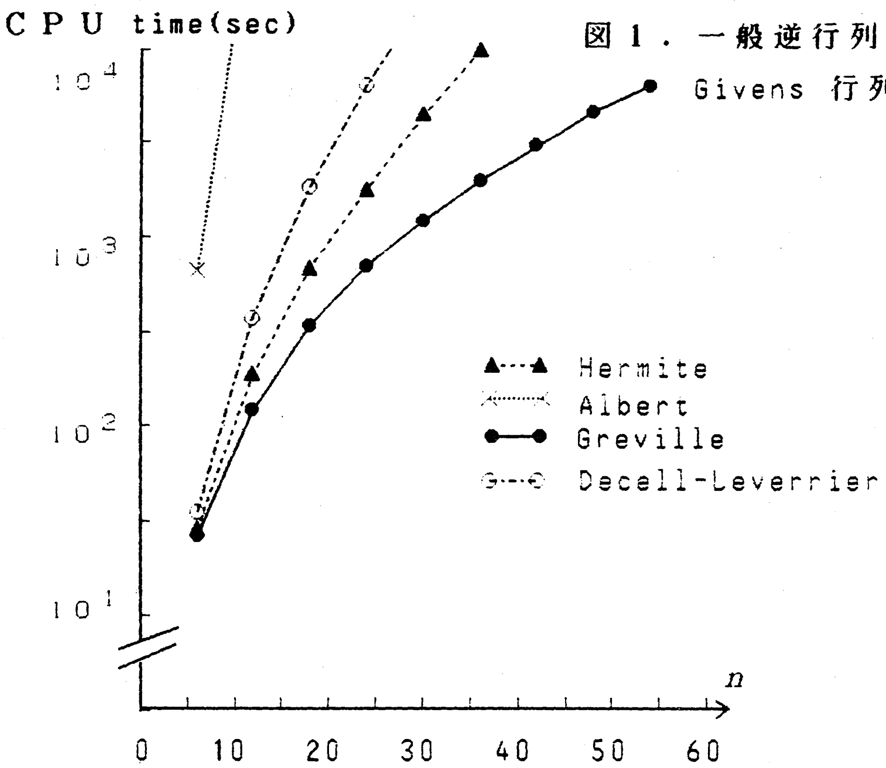

図1. 一般逆行列の計算時間

Givens 行列

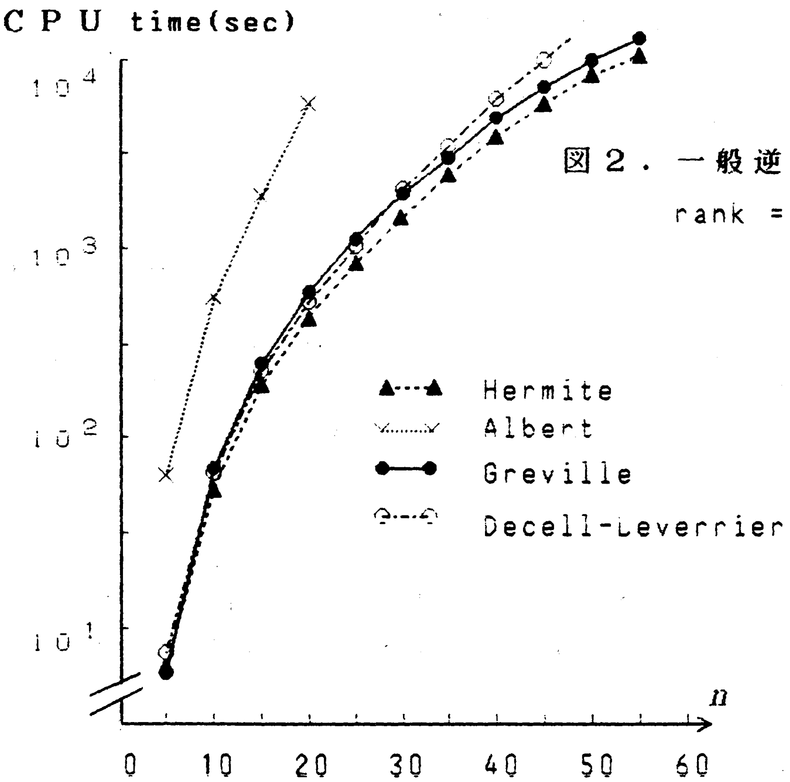

図2. 一般逆行列の計算時間

rank = 2 の行列